PreProcessingOnIrisDataset

September 22, 2025

[1]:

```python
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn import datasets
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
```

[2]:

```python
# Load the iris dataset
iris = datasets.load_iris()
iris_df = pd.DataFrame(data=iris.data, columns=iris.feature_names)
iris_df['species'] = iris.target
```

[3]:

```python
# Check whether all the attributes are standardized
print("Mean of each attribute before standardization:")
print(iris_df.iloc[:, :-1].mean())
print("\nStandard deviation of each attribute before standardization:")
print(iris_df.iloc[:, :-1].std())
```

```
Mean of each attribute before standardization:
sepal length (cm)    5.843333
sepal width (cm)     3.057333
petal length (cm)    3.758000
petal width (cm)     1.199333
dtype: float64

Standard deviation of each attribute before standardization:
sepal length (cm)    0.828066
sepal width (cm)     0.435866
petal length (cm)    1.765298
petal width (cm)     0.762238
dtype: float64
```

[4]:

```python
# Standardize the attributes if they are not standardized
scaler = StandardScaler()
iris_df.iloc[:, :-1] = scaler.fit_transform(iris_df.iloc[:, :-1])

print("\nMean of each attribute after standardization:")
print(iris_df.iloc[:, :-1].mean())
print("\nStandard deviation of each attribute after standardization:")
print(iris_df.iloc[:, :-1].std())
```

```
Mean of each attribute after standardization:
sepal length (cm)   -1.690315e-15
sepal width (cm)    -1.842970e-15
petal length (cm)   -1.698641e-15
petal width (cm)    -1.409243e-15
dtype: float64

Standard deviation of each attribute after standardization:
sepal length (cm)    1.00335
sepal width (cm)     1.00335
petal length (cm)    1.00335
petal width (cm)     1.00335
dtype: float64
```

[5]:

```python
# Aggregation
# Create a new dataset with the mean of the attributes for each species
mean_iris = iris_df.groupby('species').mean()
print("\nMean of attributes for each species:")
print(mean_iris)
```

```
Mean of attributes for each species:
         sepal length (cm)  sepal width (cm)  petal length (cm)  \
species
0                -1.014579          0.853263          -1.304987
1                 0.112282         -0.661432           0.285324
2                 0.902297         -0.191831           1.019663

         petal width (cm)
species
0               -1.254893
1                0.166734
2                1.088159
```

[6]:

```python
# Create a new dataset with the sum of the attributes for each species
sum_iris = iris_df.groupby('species').sum()
print("\nSum of attributes for each species:")
print(sum_iris)
```

```
Sum of attributes for each species:
         sepal length (cm)  sepal width (cm)  petal length (cm)  \
species
0               -50.728948         42.663134         -65.249366
1                 5.614111        -33.071602          14.266194
```

```
2                45.114837         -9.591532          50.983172

         petal width (cm)
species
0              -62.744675
1                8.336705
2               54.407970
```

[7]:

```python
# Create a new dataset with the standard deviation of the attributes for each species
std_iris = iris_df.groupby('species').std()
print("\nStandard deviation of attributes for each species:")
print(std_iris)
```

```
Standard deviation of attributes for each species:
         sepal length (cm)  sepal width (cm)  petal length (cm)  \
species
0                 0.427104          0.872594           0.098706
1                 0.625434          0.722354           0.267085
2                 0.770482          0.742377           0.313683

         petal width (cm)
species
0                0.138721
1                0.260306
2                0.361528
```

[8]:

```python
# Randomly sample 80% of the records in the Iris dataset to create a new dataset Train_iris
train_iris = iris_df.sample(frac=0.8, random_state=42)
print("\nTrain dataset (80% of records):")
print(train_iris)
```

```
Train dataset (80% of records):
     sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)  \
73            0.310998         -0.592373           0.535409          0.000878
18           -0.173674          1.709595          -1.169714         -1.183812
118           2.249683         -1.052767           1.785832          1.448832
78            0.189830         -0.362176           0.421734          0.395774
76            1.159173         -0.592373           0.592246          0.264142
..                 …                 …                  …                 …
139           1.280340          0.098217           0.933271          1.185567
61            0.068662         -0.131979           0.251221          0.395774
147           0.795669         -0.131979           0.819596          1.053935
79           -0.173674         -1.052767          -0.146641         -0.262387
59           -0.779513         -0.822570           0.080709          0.264142
```

```
     species
73         1
18         0
118        2
78         1
76         1
..       ...
139        2
61         1
147        2
79         1
59         1

[120 rows x 5 columns]
```

[9]:

```python
# Discretize Petal.length and Petal.width into three categories each named "low", "medium" and "high"
bins_length = [0, 1.5, 4.5, 7.0]  # Adjust these values based on the range of Petal.length
labels_length = ['low', 'medium', 'high']
iris_df['Petal.length.category'] = pd.cut(iris_df['petal length (cm)'], bins=bins_length, labels=labels_length)

bins_width = [0, 0.5, 1.5, 2.5]  # Adjust these values based on the range of Petal.width
labels_width = ['low', 'medium', 'high']
iris_df['Petal.width.category'] = pd.cut(iris_df['petal width (cm)'], bins=bins_width, labels=labels_width)

print("\nIris dataset with discretized Petal.length and Petal.width:")
print(iris_df[['petal length (cm)', 'Petal.length.category', 'petal width (cm)', 'Petal.width.category']])
```

```
Iris dataset with discretized Petal.length and Petal.width:
     petal length (cm) Petal.length.category  petal width (cm)  \
0            -1.340227                   NaN         -1.315444
1            -1.340227                   NaN         -1.315444
2            -1.397064                   NaN         -1.315444
3            -1.283389                   NaN         -1.315444
4            -1.340227                   NaN         -1.315444
..                 ...                   ...               ...
145           0.819596                   low          1.448832
146           0.705921                   low          0.922303
147           0.819596                   low          1.053935
148           0.933271                   low          1.448832
149           0.762758                   low          0.790671
```

```
    Petal.width.category
0                    NaN
1                    NaN
2                    NaN
3                    NaN
4                    NaN
..                   …
145               medium
146               medium
147               medium
148               medium
149               medium

[150 rows x 4 columns]
```

```
[ ]:
```